International Journal of Innovative Research in Science and Technology
Volume 01, Issue 01, August 2021
www.ijirst.com

Scenario-Based Simulation Experiments and Metrics for Certificate-Based Authentication in MANETS

P. Manoj Kumar[1], R. Jayalakshmi[2]
[1,2](Department of EEE, PSCMR CET, Vijayawada, Andhra Pradesh, INDIA)
Corresponding Author: pmanoj249@gmail.com

To Cite this Article
P. Manoj Kumar, R. Jayalakshmi, "Scenario-Based Simulation Experiments and Metrics for Certificate-Based Authentication in MANETS", International Journal of Innovative Research in Science and Technology, Vol. 01, Issue 01, August 2021, pp.:047-053.

Abstract: *Certificate-based authentication is well-studied in wired networks. Adapting certificate authentication methods for motive ad hoc networks (MANETs) is a difficult task, owing to the lack of centralized administration and infrastructure in MANETs compared to conventional wired networks. A conventional certificate-based authentication system, for example, utilizes a single trustworthy Certificate Authority (CA) to create, distribute, renew, and revoke certificates. Due to issues such as node variety, limited wireless media, and frequent connection failures, it is usually not possible to include such a fixed unified CA in the MANET approach. There are many approaches to solving the issue of using certificate-based techniques for remote authentication on mobile ad hoc networks. In this article, we make two contributions. We examine the requirements of a secured distributed authentication method for MANETs first, and then examine the characteristics of some of the existing certificate-based authentication systems in the context of distributed authentication, including their advantages and disadvantages. Finally, to evaluate these features, a set of modeling tests and metrics on the scenario were suggested.*

Keywords*: Ad hoc networks and cameras, authentication, measurement, emulation*

I. Introduction

Mobile Ad hoc networks (MANETs) have gotten a lot of interest lately, thanks in part to the fact that they may be used in a variety of applications. However, due of the complex presence of nodes, the random topology, the limited wireless range of nodes, and communication failures, the utilization of these networks poses numerous difficult issues. The wireless channel is susceptible to active and passive attacks by malicious nodes, such as service denial, eavesdropping, spot-spoofing, and so on, since all nodes in the network work together to transmit information. As a result, the encryption architecture of these networks is critical.

The five components of a protective system are confidentiality, honesty, authenticity, availability, and non-reputability. As a result, authenticity is the most important issue, since a breach of authenticity leads to a system-wide compromise. In conventional wired networks, one of the most often used authentication techniques is the public key management system using certificates.

One of the main issues with a certificate-based system is the consistent distribution of public keys to all nodes in the network. The PKI [1] specifies X.509 certificate-based public key management methods. In a wired network, a centralized certificate server is responsible for creating, updating, and revoking certificates. When there is no established structure and unified governance in ad hoc networks, this isn't required. Furthermore, owing to complicated network architecture, recurring connection failures may occur, leading to problems such as authentication and timely communication with the certificate server.

To solve these flaws and make the most of the certificate authentication process, many techniques for public key management were developed [2]. In this post, we look at a few of these methods and discuss their advantages and disadvantages. The rest of the document is organized in the same way. The requirements for a portable ad hoc network certificate-based authentication system are outlined in Section 2. The third section contains an assessment and a brief description of the methods used. Section 4 compares and contrasts the schemes and specifications. We provide scenarios and techniques for analyzing these processes via simulation in Section 5.

II. Requirements for successful certificate-based ad hoc network authentication

With any certificate-based authentication system, five requirements have been developed to guarantee secure and efficient authentication in a mobile ad hoc network.

R.1 Disseminated authentication: On ad hoc networks, it is not usually feasible to establish a fixed centralized CA due to issues such as frequent connection failures, node mobility, and limited wireless medium. Furthermore, in networks that need strong security, a server may become a single point of risk. Consider a combat scenario in which troops are dispersed over a large area. A central server may not be viable in this situation. Consider a server assault by an enemy - the whole network will go down! The main limitation of a certificate-based method is to distribute authorisation over a large number of nodes inside the network.

R.2 Resource sensitivity: Because ad hoc network nodes are often powered by batteries with limited memory capacity, authentication methods must be resource-aware. This guarantees that the underlying algorithms have a suitable time and space complexity. In this regard, symmetric-key-based encryption methods are preferable to public-key encryption strategies since symmetric encryption generally uses less energy. However, the difficulty of exchanging symmetric keys prevents them from being used in ad hoc networks. At the implementation stage, there is a compromise that has to be addressed. Because certificate-based authentication relies on resource-intensive public key methods, the protocol's memory and processing capacity must be sufficient.

R.3 Effective certificate management mechanism: the transmission of public keys and certificate management in wired networks has been extensively researched [3]. However, enforcing these techniques for MANETs is challenging due to the administration of certificates (creation, revocation, and renewal). Parts 3 and 4 go further into this. Many of the suggested procedures do not have a strict certificate revocation mechanism.

R.4. Diverse registration: As with wired networks, certifying authority in ad hoc networks are often heterogeneous. This guarantees that two or more nodes belonging to different "domains" try to authenticate each other. The certifying authority must have some kind of confidence structure or hierarchy in such a scenario. Credential chaining is used in wired networks to do this.

R.5. Pre-authentication robust mechanism: this is the technique by which the required trust is established between the nodes prior to the creation and distribution of the certificate. Although this is not part of the certificate verification process, it is critical in MANETs. This is because, in order to satisfy R.1, nodes must have previous trust in each other (by exchange of public keys, for example). Without this, the following reciprocal authentication and certificate renewal would be impossible. One of the first attempts in this area was Stajano and Anderson's [8] Resurrecting Duckling model, which featured the bootstrapped trust between a "mother" and a "pushing-in" node via a local channel. Balkans et al. [9] provide a solution that is both accessible and user-friendly. The scope of this article is too narrow to provide a comprehensive categorization of these processes.

III. Examining Related Work

Three stages are usually involved in certificate-based authentication. During the initial phase, or "bootstrapping" level, a certifying authority issues a certificate to the nodes. The certificate is created by the CA and contains the node's identifying information, such as its IP address, name, organization, and public key. The certificate includes the problem period and the expiration term, among other things. Prior to its expiration, the credential will be "renewed" during the second procedure. The CA certificate is revoked in the third step, most often by compromising the certificate holder's private key or by believing the issuer that the user's key obligation is no longer valid in some manner. We're currently looking at some of the approaches that have been mentioned.

Management of public keys in a self-contained manner:

The construction of graphs [4] is one of the authentication methods proposed by Catkin, Bettina, and Hubbub on the basis of certificates. The suggested method is similar to PGP certificates [10], with the exception that PGP uses a central credential server. A graph is defined as G (V, E), where V and E denote the set of vertices and edges, respectively. The edges of the certificate graph are public keys, while the vertices are certificates. As illustrated in Figure 1, the certificate provided by u to v, which is signed with a private key against the Kava public key, is reflected by a directed edge of the graph from Ku to Kava. In fact, the CA for v is u. Only valid network certificates are stored in G.

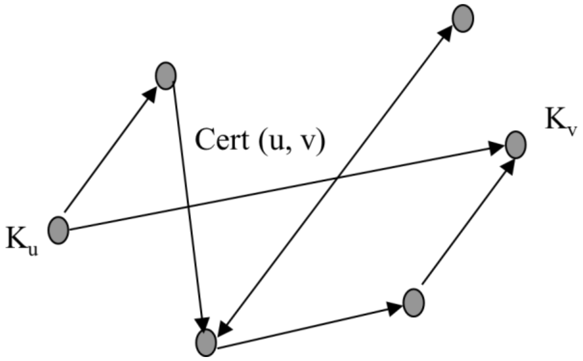


Figure no 1: Ku →KV, certificate issued to v by u

Each node keeps a current and un-updated local certificate repository, which includes a portion of updated and expired certificates. According to Catkin et al., two libraries should be utilized to give a good assessment of the certificate graph and node authentication. When a user u wants to verify the authenticity of another user's public key v, u uses updated certificate registry graphs u and v to try to locate a guided path through the network. The path is used to balance the chain of certificates in order to authenticate v. If no path is found, the node combines the un-updated and updated certificate repositories to locate the path's expired certificates. The expired certificate is updated when such a route is discovered, and the accuracy is checked and certified.

The certificate development procedure begins with each node generating its own public-private key pairs. If a new node requests a fresh certificate from its neighbor, the issuer validates the public key's authenticity. The key is pre-exchanged via a side channel, according to Catkin et al. A certificate exchange procedure is performed on a regular basis to update the certificate graphs in the updated registry by sharing certificate hashes with nearby nodes. Until all nodes have been updated with the certificate maps, the convergence times are upper limit. To maximize the utility of the modified certificate registry creation and upgrading, Catkin et al. propose methods such as Maximum Grade algorithms based on the route in certificate charts with the greatest number of certificates.

When a node discovers expired certificates in its flooded certificate store, Catkin et al. may not mention any particular technique of certificate renewal. They propose two methods for cancelling certificates: one that is explicit and the other that is oblique. On the basis of their expiry term, the certificates are removed via the tacit procedure. The issuer sends a cancelation statement to the target node, indicating that it no longer believes a valid user-key binding exists. This is delivered to nodes that are requesting certificate updates from the issuer for the target node.

The advantage of this approach is that it allows for full self-organization of public key management through certificates. The system's drawbacks include the expensive tables that credential servers must maintain, since every time a node switches locations, it must renegotiate with other nodes and update the databases.

Providing MANETs with Robust and Universal Security Support:

In this system, Kong et al [4] are exploring a distributed credential that focuses on thresholds and mutual secrets. The fundamental goal of a secret threshold sharing method is to use a secret polynomial f to exchange a secret key k with an arbitrarily large population (x). If f(x) is (k-1), any k group member may recover the hidden key, but members who are less than k will not share any information about the secret [6]. As a result, a node's public key is derived from its k neighboring nodes. In this case, k is a parameter that must be carefully tweaked in order for the procedure to work.

The credential creation process is as follows: all nodes in the network must first be booted with their certificates by a trustworthy central management. If a new node requires a certificate, it submits a request for partial certificates to its k-neighboring nodes. When the coalition decides that the requested node is well-preserved, it issues partial certificates, which are then combined by the aim of a new certificate with an interpolation feature.

A Period Renew renewal is used to renew the certificate. In a single hop, a network agency broadcasts the current valid certificate to its neighbors, as well as a possible phrase T (current time + renew), which may be used to update a certificate. To determine whether the request is allowed or refused, the nearby nodes look up the public device key and the credential revocation list.

As suggested in [2] revocation of the certification is carried out via tacit or formal procedures using two methods. If the expiration term (Expire) is less than the expiry time + the renewal time, the certificates are revoked using the tacit technique (Renew). In the transparent certificate revocation protocol, each node keeps a certificate revocation list containing certificates that have not yet expired. The node checks its CRL for expired certificates on a regular basis and, if necessary, revokes them.

The main advantage of this method is that it eliminates the requirement for a centralized certifying authority. It does, however, need at least k one-hop authentication neighbors. Because of the complicated existence of the nodes, this is not possible if k is large. Furthermore, certificates cannot be released with nodes that are more than one hop apart. It also requires a bootstrapping procedure to distribute the system's private key across the k nodes at first.

Heterogeneous Certification that is Self-Managed:

Wang, Zhu, and Li [3] provide a novel paradigm for the coexistence of CAs from different administrative regions in the network. They also propose a distributed certificate authority based on a concealed k-threshold exchange, similar to the Kong et al approach [4]. Heterogeneous CAs are treated using Esteem diagrams. If node B can be authenticated using the digital B certificate issued by the CA that A presently trusts, node A is said to have trusted it. A list of trustworthy CAs is kept on each node.

Whenever a node wants a credential, it must collect K IDs from its valid shareholder one-hop neighbors and create a private key. When a node needs to authenticate another node, it begins by providing B with its CA list. Similarly, B provides a CA list of its own. Then A checks the two lists to see whether there are any common CAs, and if there are, A transmits its certificate to B, who is also CA-certified. B replies by presenting A with a certificate of its own. If the two nodes share a CA, they verify their one-hop and two-hop neighbors using the distributed multi-hop certificate request (DMCR) method.

The stages of certificate renewal are similar to those of the DMCR system. The issue of license rejection, on the other hand, is not addressed.

The main advantages of this approach are: I cross-certification help between ACs in a variety of regions; and (ii) the certificate finding procedure is carried out in a variety of stores.

Authentication based on trust and clustering:

In order to improve public certification security, Ngami et al [5] address a faith model and a network model. Your network design is centered on the network's hierarchical structure or clustering through clustering techniques. Such methods, according to the authors, improve network security and performance. You think the network is divided into clusters with unique names.

Their confidence mechanism is built on a web-of-trust concept similar to PGP [10], in which each person acts as a certifying authority. They express confidence in terms of a constant number between 0 and 1. A list of trust values for other network nodes is kept by each node. A direct trust is defined as a trust connection between two nodes in the same community, while a confidence of suggestion is defined as a trust relationship between nodes of different classes. The nodes are anticipated to be equipped with tracking components, such as a surveillance dog for node activities, in order to establish the trust connection.

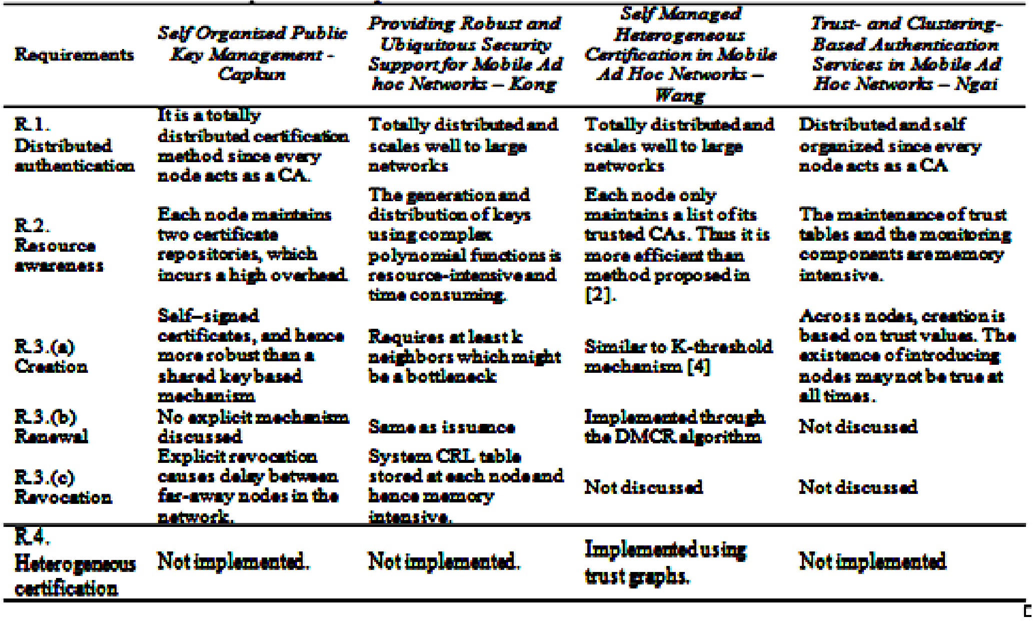
It's thought that public primary control exists inside a cluster. If a node in one cluster wants to authenticate a node in another, it must connect to a large number of other nodes in the cluster. It ranks the introductory nodes according to their trust values and calculates their weighted trust value by comparing the introductory nodes' trust values with the introductory nodes' trust values to the objective node. The confidence's final value is then stored and utilized to compare other nodes.

The authors don't go into detail on the procedure of renewal and cancellation. In comparison to PGP-based methods, the process has the disadvantage of detecting and isolating a large proportion of rogue nodes. The disadvantage is that storing and measuring trust levels requires memory and time. Furthermore, node mobility leads to changes in node membership across different clusters.

IV. Mechanisms are compared

Table 1 compares the four mechanisms to the above-mentioned criteria. We may choose not to examine R.5 since it is not a necessary part of the credentialing process.

Table no 1: Comparison of Certificate-based Authentications



Requirements	*Self Organised Public Key Management - Capkun*	*Providing Robust and Ubiquitous Security Support for Mobile Ad hoc Networks – Kong*	*Self Managed Heterogeneous Certification in Mobile Ad Hoc Networks – Wang*	*Trust- and Clustering-Based Authentication Services in Mobile Ad Hoc Networks – Ngai*
R.1. Distributed authentication	It is a totally distributed certification method since every node acts as a CA.	Totally distributed and scales well to large networks	Totally distributed and scales well to large networks	Distributed and self organized since every node acts as a CA
R.2. Resource awareness	Each node maintains two certificate repositories, which incurs a high overhead.	The generation and distribution of keys using complex polynomial functions is resource-intensive and time consuming.	Each node only maintains a list of its trusted CAs. Thus it is more efficient than method proposed in [2].	The maintenance of trust tables and the monitoring components are memory intensive.
R.3.(a) Creation	Self–signed certificates, and hence more robust than a shared key based mechanism	Requires at least k neighbors which might be a bottleneck	Similar to K-threshold mechanism [4]	Across nodes, creation is based on trust values. The existence of introducing nodes may not be true at all times.
R.3.(b) Renewal	No explicit mechanism discussed	Same as issuance	Implemented through the DMCR algorithm	Not discussed
R.3.(c) Revocation	Explicit revocation causes delay between far-away nodes in the network.	System CRL table stored at each node and hence memory intensive.	Not discussed	Not discussed
R.4. Heterogeneous certification	Not implemented.	Not implemented.	Implemented using trust graphs.	Not implemented

V. Metrics and Scenarios

We provide a range of realistic simulation scenarios to investigate the viability of these procedures. Before we can decide on scenarios, we must first establish specific criteria.

Parameters for defining scenarios:

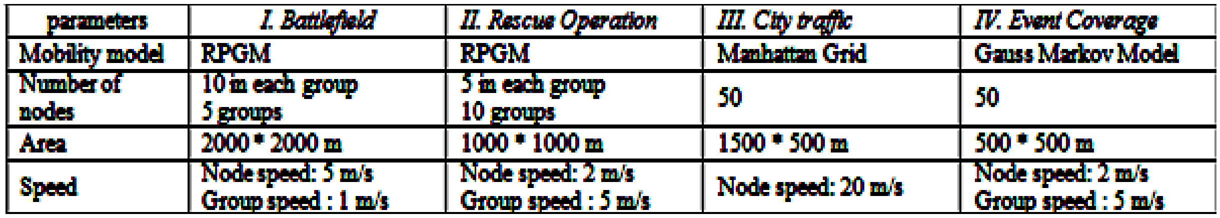
1) The mobility model is based on real-world network node movement. Mobility models for companies and mobility models for classes are the two main categories. These models are classified more broadly by Camp et al. [11]. The RWM (Random Waypoint Model), which utilizes stop intervals and random changes in destinations and speed, is the most common mobility model among scientists. As a result, randomness does not suit well in certain circumstances, such as a combat zone, where movement is more predictable. Furthermore, even after a lengthy simulation period, the model fails to guarantee "statelessness"[12]. As a result, when testing and authentication methods that are reliant on the credential, mobility models should be carefully chosen. The actual scenario should be as precise as feasible.

2) The density of nodes varies greatly depending on the situation. For example, an incident coverage scenario could have a high node density, while a disaster recovery scenario might have a low density since nodes are spread out across a large area.

3) Traffic levels vary depending on node connection failures, congestion, and mobility. As the scenario is defined, the sources and traffic kind (e.g., CBR, TCP, or UDP) must also be handled. The most common traffic format is constant bit rate (CBR). A packet rate and size of 4 packets per second and 512 bytes, respectively, may be used as an example.

Table 2 lists examples of circumstances and the simulation settings that go with them. Scenarios I and II are based on the Reference Point Group's mobility model (RPGM) [11]. RPGM is a model of community mobility in which each group has a rational center (like a soldier's head) that determines collective behavior. The RWM moves the nodes inside a community at random, but the leader controls the Group's overall mobility. Scenarios III and IV are concerned with agent mobility templates. The Random Waypoint object versatility model is the most frequently used. For realistic scenarios, the Manhattan Grid Concept is utilized in scenario III, while the Gauss Markov Model is featured in scenario IV.

Table no 2: Sample Scenarios



parameters	I. Battlefield	II. Rescue Operation	III. City traffic	IV. Event Coverage
Mobility model	RPGM	RPGM	Manhattan Grid	Gauss Markov Model
Number of nodes	10 in each group 5 groups	5 in each group 10 groups	50	50
Area	2000 * 2000 m	1000 * 1000 m	1500 * 500 m	500 * 500 m
Speed	Node speed: 5 m/s Group speed : 1 m/s	Node speed: 2 m/s Group speed : 5 m/s	Node speed: 20 m/s	Node speed: 2 m/s Group speed : 5 m/s

Metrics:

We created the following measurements after specifying the scenario settings, which may be used to test authentication methods. [7] was used to modify any of the metrics.

The number of successful certification services (including issuance, NCISS, and renewal, NCREN, respectively) to the total number of requests for such services is measured by the Successful Certification Ratio (). (NCTOT-ISS and NCTOT-REN, respectively). It provides an indication of the mechanism's effectiveness in delivering successful certification services. If we use REN to represent the successful certification renewal ratio and ISS to represent the successful certificate issuance ratio, we can compute their respective values as follows:

$$\mu_{REN} = \frac{NC_{REN}}{NC_{TOT-REN}} \qquad \mu_{ISS} = \frac{NC_{ISS}}{NC_{TOT-ISS}}$$

NCREN and NCISS are the total number of certificates renewed and issued, respectively, while NCTOT-REN and NCTOT-ISS are the total number of certificate issuance and renewal requests, respectively.

a) Settling time (st) is the time it takes for all nodes in a network to get valid certificates for the first time. The difference between the time when all nodes are given valid certificates and the time when the process of certificate issuance starts may be used to determine the value of st. The amount of time it takes to settle will be determined by a variety of variables, including the number of malicious or non-cooperative nodes, the methods employed for key creation and distribution, and so on. The settling time will be shorter if the pre-authentication procedures are efficient (R.5).

Frequency of Certification (fcert) measures the number of certification services per time interval.

$$f_{cert} = \frac{N_{cert}}{T_{int}}$$

Here Ncert is the total number of certification services (issuance/renewal) by nodes in the network, and Tint is the simulation time. As the topology of the As the network evolves, it is anticipated that certificate issuing and renewal procedures would become more frequent. This adds overhead because each time a node wants to generate or renew its certificate, expensive calculations for the public key mechanism must be performed. We expect a distributed and self-organized process to have a lower fcert and therefore a lower frequency of certificate generation, renewal, and revocation.

b) The average certification delay (acd) is calculated by averaging the time delay between the certificate service request (CSReq) and the certificate service reply (CSRep) throughout the simulation period.

$$acd = \frac{\Sigma_{i=1..n}(CS\,\mathrm{Re}\,p_i - CS\,\mathrm{Re}\,q_i)}{T_{int}}$$

This number assesses the algorithm's efficiency and is mostly determined by the algorithm's time complexity.

VI. Conclusion and Future Work

In mobile ad hoc networks, successful authentication is critical for ensuring that the sponsored application operates securely and effectively, particularly in distant field applications when mobile nodes are spread over a large geographic area. Several certificate-based security methods for MANETs have been proposed. Some of these procedures are examined, and certificate authentication requirements for MANETs are defined. We also provide a number of theoretical scenarios and data that may be utilized in Network Simulator ns-2 simulation studies.

References

[1]. Internet X.509 Public Key Infrastructure Certificate and CRL Profile - RFC 2459.

[2]. S. Capkun, L. Buttyan and J-P Hubaux. "Self-Organized Public-Key Management for Mobile Ad Hoc Networks ", IEEE Transactions on Mobile Computing, Vol. 7, No. 1, Jan-Mar 2010, pp. 52-64

[3]. Weihong Wang, Ying Zhu, Baochun Li. "Self-Managed Heterogeneous Certification in Mobile Ad Hoc Networks ", in the Proceedings of IEEE Vehicular Technology Conference (VTC 2003), Orlando, Florida, 10/6-9, 2011.

[4]. J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang. "Providing robust and Ubiquitous Security support for Mobile Ad Hoc Networks ", Proceedings of the 9th International conference on Network Protocols (ICNP), Riverside, California, USA, November 11-14 2012.

[5]. Edith C. H. Ngai and Michael R. Lyu. "Trust- and Clustering-Based Authentication Services in Mobile Ad Hoc Networks", 24th International Conference on Distributed Computing Systems Workshops - W4: MDC (ICDCSW'04), Hachioji, Tokyo, Japan, 3/23-24, 2004.

[6]. L. Zhou and Z. Haas. “Securing Ad Hoc Networks”, IEEE Network magazine, special issue on networking security, Vol. 13, No. 6, November/December 1999.

[7]. Matei Ciobanu Morogan, Sead Muftic. “Certificate Management in Ad Hoc Networks”, 2003 Symposium on Applications and the Internet Workshops (SAINT'03 Workshops), January 27 - 31, 2003, pp. 337.

[8]. F. Stajano and R. J. Anderson. “The resurrecting duckling: Security issues for ad-hoc wireless networks” In 7th Security Protocols Workshop, volume 1796 of Lecture Notes in Computer Science, Cambridge, United Kingdom, 1999. Springer-Verlag, Berlin Germany.

[9]. Dirk Balfanz, D. K. Smetters, Paul Stewart and H. Chi Wong: "Talking To Strangers: Authentication in Ad-Hoc Wireless Networks", Symposium on Network and Distributed Systems Security (NDSS'02), Xerox Palo Alto Research Center, Palo Alto, USA, 2002.

[10]. P. Zimmerman. The Official PGP Users guide, MIT Press, 1995, ISBN 0-262-74017-6.

[11]. T. Camp, J. Boleng, and V. Davies. “A Survey of Mobility Models for Ad Hoc Network Research", in Wireless Communication & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, vol. 2, no. 5, 2002.

[12]. J. Yoon, M. Liu, and B. Noble. "Random waypoint considered harmful," in Proc. of IEEE INFOCOM '03, vol. 2, March 2003, pp. 1312—1321.

[13]. K. Fall and K. Varadhan, the NS Manual, the VINT Project, UC Berkeley, January 2002.